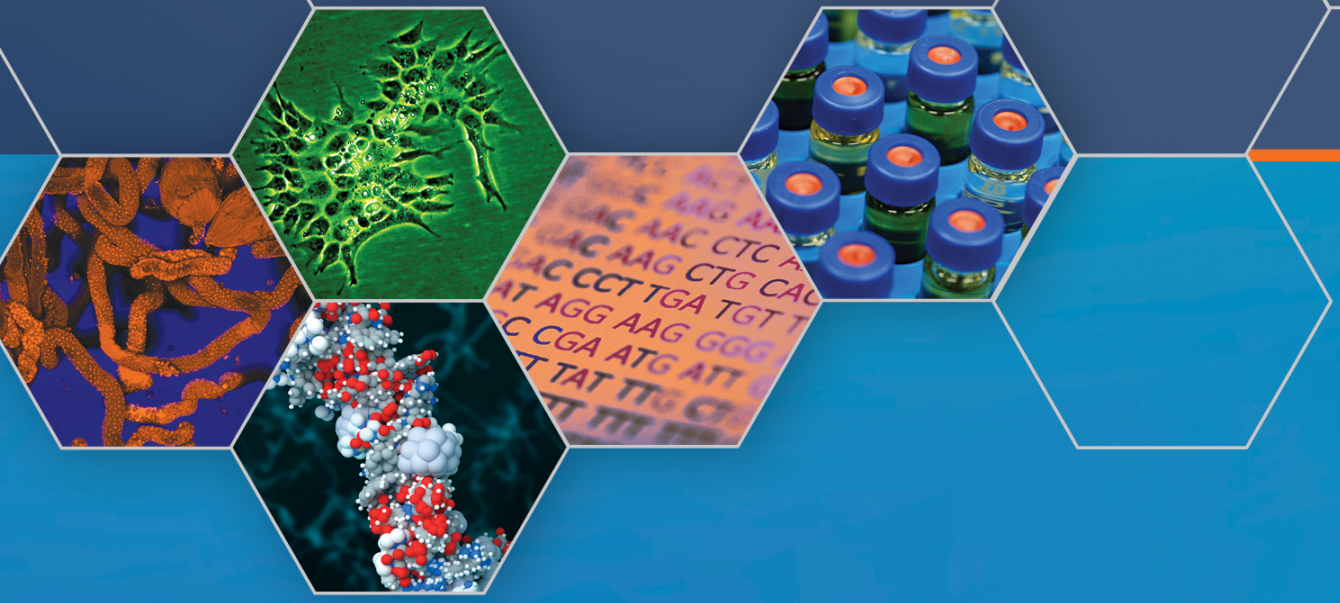
ICBR

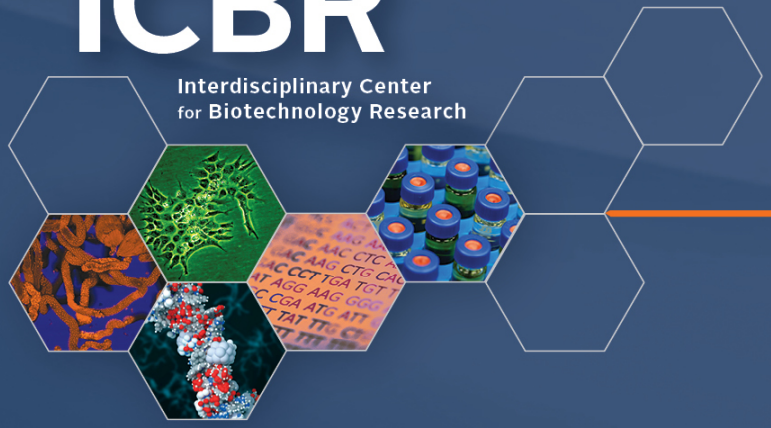
Interdisciplinary Center
for Biotechnology Research



UF | UNIVERSITY of FLORIDA

ICBR

Interdisciplinary Center for Biotechnology Research



Bioinformatics 101 – Lecture 5

RNA-Seq Analysis

Alberto Riva (ariva@ufl.edu), J. Lucas Boatwright (jlboat@ufl.edu)

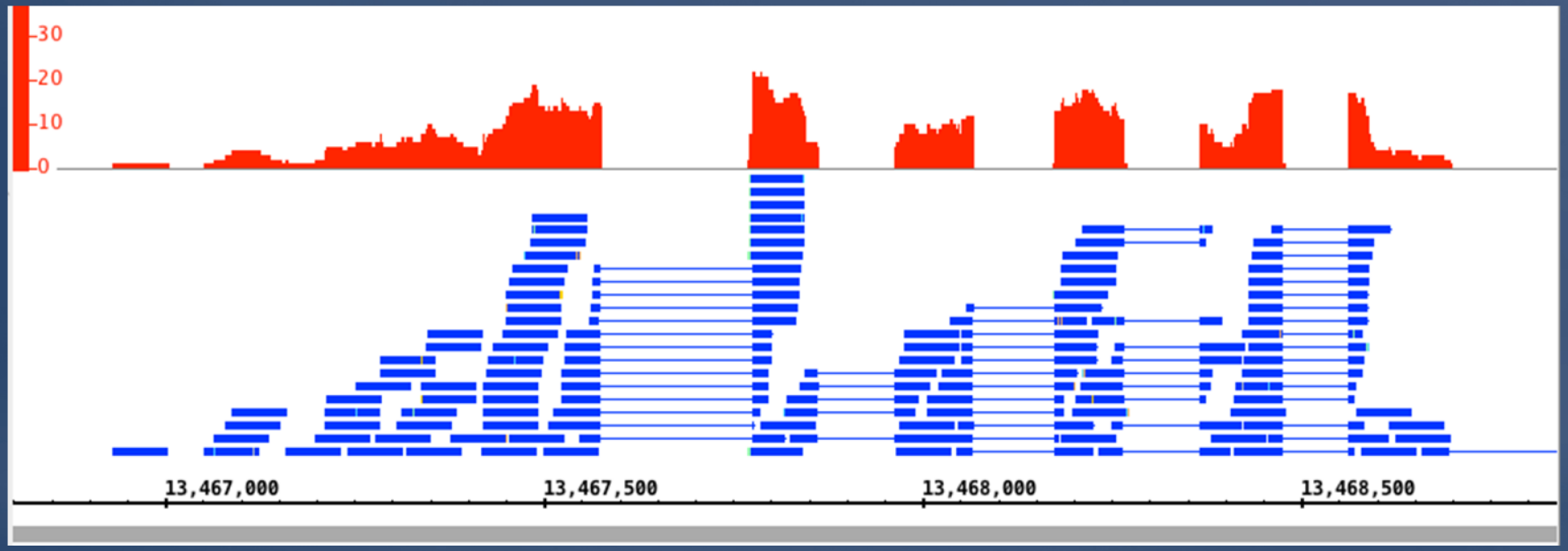
ICBR Bioinformatics Core

UF | UNIVERSITY of FLORIDA

RNA-Seq basics

- Sequencing can be applied to mRNA (expressed transcripts).
- The number of reads originating from each mRNA can be counted ("sequence census" methods) is assumed to be proportional to the mRNA expression.
- Digital, not continuous, measure of gene expression. Unit of measure: number of reads per kilobase per million reads (RPKM).

RNA-Seq analysis



UF | ICBR BIOINFORMATICS

UFHealth CANCER CENTER

RNA-Seq advantages

- Accuracy and reproducibility generally higher than microarrays.
- Unbiased: allows detection of previously-unknown transcripts and splice variants.
- Does not necessarily require a reference sequence: since the transcriptome is more conserved than the genome, a related genome can be used.

RNA-Seq limitations

- If coverage is low / uneven, expression values can be inaccurate (especially for low-expression transcripts);
- Transcripts expressed at very high levels can saturate the analysis (i.e., absorb the majority of reads);
- Accurate gene expression measurements (especially when comparing different samples) require a large number of biological replicates.

RNA-Seq analysis

Basic steps:

- Alignment of reads to transcriptome;
- Optionally: transcriptome reconstruction;
- Quantification: assign reads to transcripts, count, normalization;
- Differential analysis: identify transcripts showing significant change in expression.

RNA-Seq analysis

Tools:

- Alignment: STAR, HISAT2, Tophat, BWA, Bowtie2.
- Transcriptome reconstruction: StringTie, Strawberry, Trinity (de novo), Velvet (de novo).
- Quantification: RSEM, StringTie, Sailfish, Salmon, Kallisto.
- Differential analysis: DESeq2, edgeR, limma-voom, ballgown, NOISeq, baySeq, EBSeq.

Sequence alignment in NGS

Mapping: given a short read, find where it originated from in the genome.

Special case of the alignment problem, in which the query is several orders of magnitude shorter than the database.

The mapping step needs to be performed tens of millions of times → need *fast* algorithms.

Efficient mapping

At first glance, mapping is actually a relatively simple problem. Can be addressed using **hash** methods.

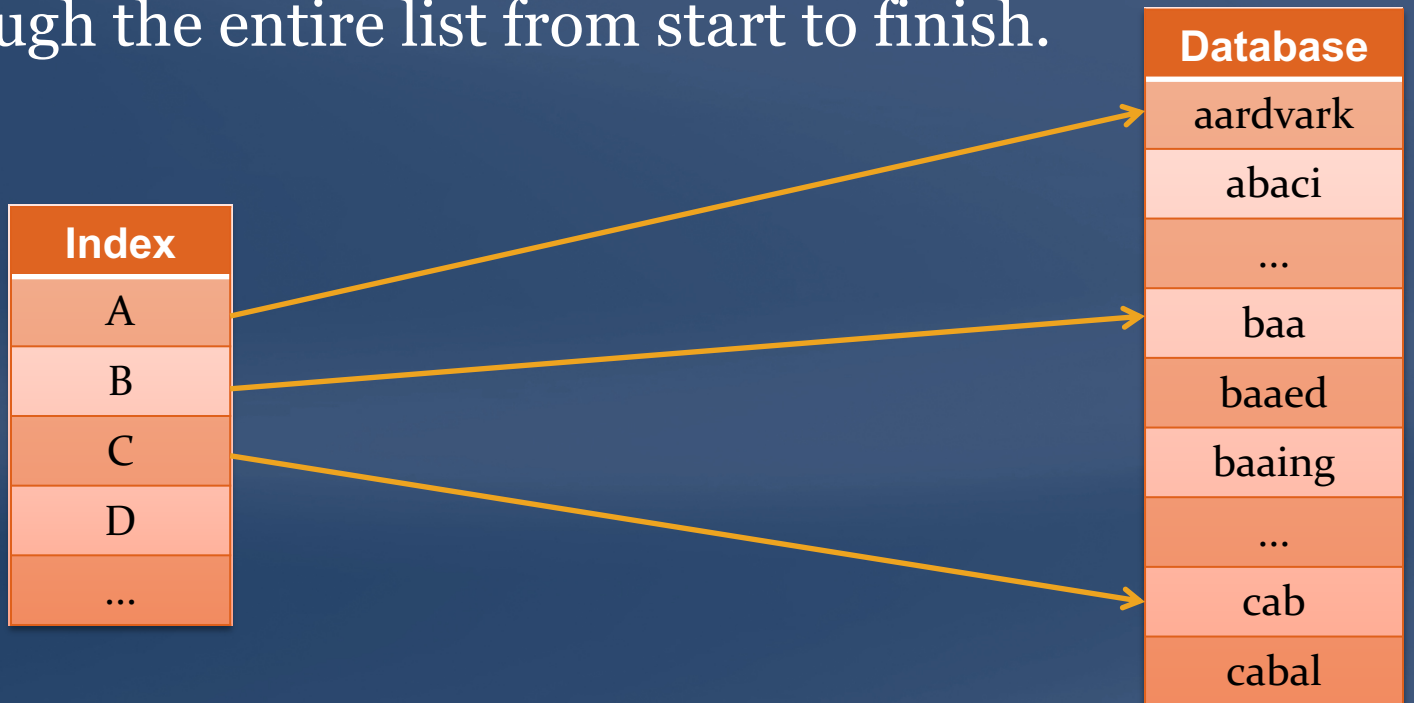
Divide the genome into “seeds”, strings of DNA of a given length.

Record all positions where a seed appears in the genome in a compressed table (**index**).

Look up portions of each read in the index.

Indexing and hashing

In computer science, an *index* is a data structure that enables sub-linear time lookup. In other words: when searching for an element in a list, you don't have to look through the entire list from start to finish.



Indexing and hashing

A **hash function** is a function that associates an object (*key*) with a *value* in a unique and efficient way.

The value can be used as a pointer in an index, to quickly retrieve data associated with the key.

Indexes and hash functions allow for fast lookups in very large lists (ideally, *constant time* access).

Hash-based mapping

Example (seed size = 4):

```
   ACGTACGTAAG
0  ACGT
1   CGTA
2    GTAC
3     TACG
4      ACGT
5       CGTA
6        GTAA
7         TAAG
```

Hash-based mapping

Group and sort:

Hash-based mapping

Each “word” is a key in the hash table. The value associated with each word is the list of positions in the genome where ot occurs.

Index creation takes time, but is only done once. The size of the index depends on:

- Word size (small: fewer keys, more occurrences; large: more keys, fewer occurrences);
- Genome size (larger: more positions for each key).

Hash-based mapping

Search strategy: break read into seed-sized fragments, and look them up in index.

Almost exact matches wanted. Need to be able to deal with mismatches, insertions and deletions.

Hash index needs to be **compressed** so that it can be stored in memory for maximum search speed. The most widely used technique in current alignment programs is the Burrows-Wheeler Transform (BWT).

Alignment software

Name	Technique	Developer
Eland	Hash of reads	Illumina
SOAP	Hash of reference	BGI
Maq	Hash of reads	Sanger
Bowtie2 / Tophat	BWT	Salzberg/UMD
BWA	BWT	Sanger
SOAP2	BWT + Hashing	BGI
STAR	Suffix arrays	CSHL

Alignment results

Alignment results are typically represented as BAM files. Features:

- Compressed and indexed;
- Contain full information on each read (sequence, quality scores);
- Can contain aligned, unaligned, partially and multiply-aligned reads.
- Describes details of alignment (gaps, mismatches);
- Additional information included through custom tags.

Alignment results

Alignment results are typically represented as BAM files. Features:

- Compressed